



次の I さんと A さんの会話文を読み、問いに答えよ。

I さん：数学の素因数分解が苦手なんだよね。「51」が素数に思えてしまうから、よくまちがえるんだ。

A さん：素数はある程度覚えておいた方が、計算が楽だね。先生は 100 以下の素数を覚えていると
 いていたけど、いったい何個あるんだろう。

I さん：えっと、素数の定義は「1 と自分以外で割れない自然数」だから、2、3、5、7、11・・・た
 くさんありすぎて考えるのが大変だ！

A さん：じゃあプログラムをつくって判断させてみようか。どうやって考えたらいいのかな？

I さん：「割り切れる」は、割った余りが 0 であるということだね。余りを求める演算子の「%」を使
 ってみよう。それで、ある自然数が素数かどうかを判定するプログラムを考えてみよう。

I さんと A さんは、変数 x に自然数を代入して、それが素数かどうかを判定するプログラムを Python
 で考えてみた。ここでは、例として x に 51 を代入した。【1】～【3】に当てはまるものを答えよ。

```
x = 51
for i in range(2, [ 1 ]):
    if [ 2 ]:
        print(x, "は素数ではない")
```

作成したプログラムを実行させると、「51 は素数ではない」と表示された。ほかに、1 行目に 20
 や 100 など代入して試してみても「素数ではない」と表示され、一見うまくいったように思えた
 が、何度も素数ではないと表示されてしまった。また、素数の 11 を代入した場合は【3】。

27

【1】の選択肢

- ① x ② x - 1 ③ i ④ i - 1

47

【2】の選択肢

- ① $i \% x == 0$ ② $i \% x != 0$ ③ $x \% i == 0$ ④ $x \% i != 0$

※「==」は「等しい」、「!=」は「等しくない」を意味する。

21

【3】の選択肢

- ① 「11 は素数ではない」と一度表示された ② 「11 は素数ではない」と何度も表示された
 ③ プログラムが動かなかった ④ 何も表示されなかった

I さん：このプログラムでも判定はできるけど、スマートではないね。もう少し別の方法で考えてみ
 ようか。

A さん：何度も「素数ではない」と表示されるのは、繰り返しの途中でストップしていないからだね。
 1 つでも割り切れてしまったら素数にはならないので、割り入れたらすぐに終了させたいな。繰り返
 しから抜けるために【6】を入れるといいね。

I さん：あと、1 つも割り切れなかったら「素数である」ことを表示させたいね。判定のための変数
 hantei に文字を代入してみよう。

I さんと A さんは、あらたに改善を加えたプログラムを考えた。ここでも x には 51 を代入した。

```
x = 51
i = 2
hantei = "true"
while i <= [ 4 ]:
    if [ 5 ]:
        hantei = "false"
    [ 6 ]
    else:
        [ 7 ]
print(hantei)
```

このプログラムを実行すると、51 の場合は「false」と表示され、11 の場合は「true」と表示された。
 また、何度も表示されることなく、一度ずつ表示された。

A さん：うまくいったね。これでどんな大きい自然数でも判定できるね。いろいろな数で実行してみ
 よう。401・・・これは素数だ！

I さん：おもしろいね。もっと大きな数で試してみよう。

38

【4】の選択肢

- ① x ② x - 1 ③ i ④ i - 1

46

【5】の選択肢

- ① $i \% x == 0$ ② $i \% x != 0$ ③ $x \% i == 0$ ④ $x \% i != 0$

※「==」は「等しい」、「!=」は「等しくない」を意味する。

25

【6】と【7】の選択肢

- ① $x = i$ ② $i = x$ ③ $i = i + 1$ ④ $x = x + 1$

[1]~[7] 2点

情報の授業で、進数計算のプログラムを作成した。授業で作成したのは 10 進数から 2 進数、2 進数から 10 進数にそれぞれ変換するプログラムだが、これを応用したら 10 進数から 8 進数に変換するプログラムが作れるのではないかと考えた。【8】～【12】に当てはまるものを答えよ。

```
jushin = int(input())
keta = 24【 8 】①
hachi = 0
while jushin > 0:
    hachi += jushin % 29【 9 】 * keta ③
    keta *= 17【 10 】 ④
    jushin //= 13【 11 】
print(46【12】) ⑥ ③
```

【8】～【12】の選択肢
① 1 ② 2 ③ 8 ④ 10 ⑤ keta ⑥ hachi ⑦ jushin

「16,23,31,48,52,65,79」の中から「31」「48」「65」を探索するとき、逐次探索と二分探索のそれぞれのアルゴリズムでの探索回数を答えよ。

データ	逐次探索での回数	二分探索での回数
31	54 ²⁶ 【 13 】 ^③	30 ³⁰ 【 16 】 ^③
48	52 ²⁷ 【 14 】 ^④	32 ³² 【 17 】 ^①
65	48 ²⁵ 【 15 】 ^⑥	28 ²⁸ 【 18 】 ^②

) 2点

```
a = [16,23,31,48,52,65,79]
i = 0
j = 26【 19 】 ⑥
s = 36
count = 0
while i<=j:
    count+=140
    m = int(27【 20 】) ③
    if list27[27【 21 】] == s: ①
        print("Hakken!",count)
        break
    if list[m] < s:
        25【 22 】 ④
    else:
        25【 23 】 ⑦
```

二分探索を左下の通りに組んでみた。
【19】に入る数値をマークせよ。
【20】～【23】に入る内容を選択せよ。
① i-j ② i+j/2 ③ (i+j)/2 ④ i = m + 1 ⑤ i = m - 1 ⑥ j = m + 1 ⑦ j = m - 1
⑧ i ⑨ j ⑩ m
左下の二分探索のプログラムにおいて、「s=52」と修正した。この時、「52」が見つかるまでの変数 i,j,m および list[m]の値を答えよ。

i の値	j の値	m の値	list[m]の値
0	6	3	48
11 ¹¹ 【24】 ^④	11 ¹¹ 【25】 ^⑥	13 ¹³ 【26】 ^⑤	18 ¹⁸ 【27】 ^{⑥5}
15 ¹⁵ 【28】 ^④	15 ¹⁵ 【29】 ^④	16 ¹⁶ 【30】 ^④	27 ²⁷ 【31】 ^{⑤2}

【27】と【31】は次から選択せよ。 ①16 ②23 ③31 ④48 ⑤52 ⑥65 ⑦79

次の5つのプログラムをそれぞれ実行した際に表示される内容をマークの番号で解答せよ。

¹⁰【32】 ^⑤
a = 3 ^② ^③
a = 5 ^② ^③
print(a)

¹⁷【33】 ^⑥
Data = [2 , 6 , 4]
print(Data[1])

⁶⁴【34】 ^②
x = 6
y = 3
z = x / y
print(z)

³⁷【35】 ^⑨
b = 7
c = 4
d = b / (^⑦c + ^④3) + 8
print(d)

⁶⁰【36】 ^⑥
e = 3
f = 2
g = f * e
print(g)

次の文章を読み、問に答えよ。
欲しいものを買うために 10,000 円を貯めたいと思ったあなたは、今日から毎日ある規則で欠かさず貯金をしていくことに決めた。貯金を開始してからの日数を x 日目とし、x が素数の場合には 100 円を、それ以外の場合は 10 円を貯金する。貯金を始めてから 6 日までの金額は、下の表のようになった。

日数	1 日目	2 日目	3 日目	4 日目	5 日目	6 日目	^⑦ ^⑧ ^⑨ ^⑩
貯金額	10	100	100	10	100	10	100 ¹⁰⁰ 10 ¹⁰ 10 ¹⁰ 10 ¹⁰
合計額	10	110	210	220	320	330	430 ⁴⁴⁰ 440 ⁴⁵⁰ 450 ⁴⁶⁰

⁷⁵【37】 9 日目の貯金額を答えよ。 ^①10 ② 100
⁵⁹【38】 10 日目の合計額を答えよ。 ① 440 ② 450 ^③460 ④ 470 ⑤ 480

```
goukei = 0
x = 0
while 【39】:
    x = x + 1
    if sosuu(x) == 1:
        goukei = goukei + 【40】
    else:
        goukei = goukei + 【41】
print(【42】)
```

```
sosuu()の例

print(sosuu(5))
実行例
1

print(sosuu(9))
実行例
0
```

貯金のプログラムを書くと、上記の通りとなった。なお、独自に宣言した関数 sosuu は、与えた値が素数であるならば 1 を返し、そうでなければ 0 を返すものである。【39】～【42】に当てはまるものを選択せよ。

16 【39】の選択肢

- ① goukei != 10000 ② goukei == 10000 ③ goukei <= 10000 ④ goukei < 10000

【40】～【42】の選択肢

- ① 1 ② 10 ③ 100 ④ x ⑤ x - 1 ⑥ x + 1 ⑦ x + 2 ⑧ goukei ⑨ goukei + x

[16,49,36,25]をソートするプログラムについて考える。以下の問に答えよ。

ソートプログラム①

```
a = [16,49,36,【44】]
for i in range(3,0,-1):
    for j in range(i):
        if a【45】 > a【46】:
            tmp = a【47】
            a【48】 = a【49】
            a【50】 = tmp
    print(a)
```

とあるアルゴリズムでソートしたら上記の通りとなった。このソートプログラム①のアルゴリズム名を答えよ。【43】

- ① マージソート ② 選択ソート ③ バブルソート ④ クイックソート ⑤ 挿入ソート

【44】～【49】に当てはまるものを答えよ

- ① j ② j + 1 ③ i ④ i + 1 ⑤ a ⑥ tmp ⑦ len(a) ⑧ len(a) - 1 ⑨ 25

ソートプログラム①を実行したとき、print(a)が 6 回実行された。その時のリスト a の中身と変数 i、変数 j の中身を答えよ。

16 49 36 25
16 36 49 25
16 36 25 49
16 36 25 49
16 25 36 49

a[0]	a[1]	a[2]	a[3]	i	j
【50】	【51】	【52】	【53】	【54】	【55】
【56】	【57】	【58】	【59】	【60】	【61】
【62】	【63】	【64】	【65】	【66】	【67】
【68】	【69】	【70】	【71】	【72】	【73】
【74】	【75】	【76】	【77】	【78】	【79】
【80】	【81】	【82】	【83】	【84】	【85】

- ① 16 ② 25 ③ 36 ④ 49 ⑤ 0 ⑥ 1 ⑦ 2 ⑧ 3 ⑨ 4 ⑩ 5

1 ~ 50 愚
51 ~ 90 知

ソートプログラム②

```
a = [16,49,36,25]
for i in range(3):
    for j in range(i+1,4):
        if a[i] > a[j]:
            tmp = a[j]
            a[j] = a[i]
            a[i] = tmp
    print(a)
```

次に、別のアルゴリズムでソートしたら上記の通りとなった。このソートプログラム②のアルゴリズム名を答えよ。【86】

- ① マージソート ② 選択ソート ③ バブルソート ④ クイックソート ⑤ 挿入ソート

このアルゴリズムでソートした際の、4 回目の print(a)で出力されたリストの内容をそれぞれ答えよ。

a[0]	a[1]	a[2]	a[3]
16	49	36	25
【87】	【88】	【89】	【90】
16	25	36	49

【87】～【90】の選択肢

- ① 16 ② 25 ③ 36 ④ 49